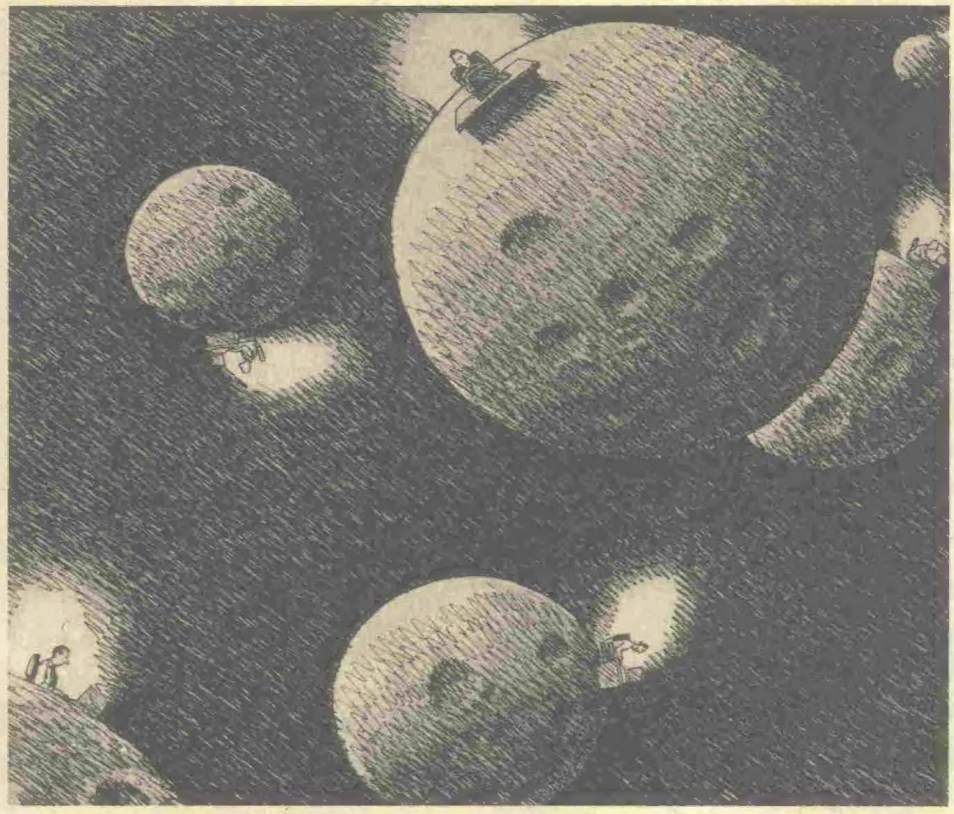
เ เ น ะ น ำ

NOVELL NetWare 2.1



ตอนจบ

สหัส พรหมสิทธิ์

ประจักษ์พยานอันเกี่ยวแก่ประสิทธิภาพของซอฟต์แวร์ที่แท้จริงนั้นจะต้องดูจากผลของการทดสอบในแง่ของการใช้งานจริง ๆ ว่า มีจุดอ่อนจุดแข็งในแต่ละด้านมากน้อยแค่ไหนนอกเหนือไปจากการรู้ว่าซอฟต์แวร์จะทำอะไรให้ได้บ้าง ดังได้กล่าวมาในสองตอนแรกของบทความเรื่องนี้ในไมโครฯ 44 และ 46 ในฉบับนี้จะกล่าวถึงการทดสอบประสิทธิผลของการใช้งานซอฟต์แวร์ของ LAN โนเวลล์เน็ตแวร์ 2.1 อย่างสมบูรณ์และขีดความสามารถของเน็ตแวร์ในส่วนที่ยังไม่ได้พูดถึงดังต่อไปนี้

การทดสอบประสิทธิภาพของเน็ตแวร์ 2.1

การวัดผล

โดยอาศัยผลของการทดสอบจากวารสาร PC Tech Journal ที่ใช้โปรแกรมในการทดสอบประสิทธิภาพของ LAN ที่ชื่อว่า LANPERF (ย่อมาจาก LAN PERFORMANCE) ซึ่งโปรแกรมนี้พัฒนาขึ้นมาวัด throughput ของการใช้งานต่าง ๆ ที่ต้องเรียกใช้ดอสอยู่ตลอดเวลา โดยมีหน่วยของ throughput คิดเป็น KB/s หรือพันไบต์ต่อวินาที

ผู้ทดสอบเบนช์มาร์คสามารถใช้ LANPERF เพื่อเปรียบเทียบประสิทธิผลของแลนซอฟต์แวร์ภายในเน็ตเวอร์กเพียงอันเดียวหรือเน็ตเวอร์กสองชนิดที่แตกต่างกันได้ภายใต้ configurations (ส่วนประกอบในด้านฮาร์ดแวร์) ของเน็ตเวอร์กที่ไม่ต่างกันออกไปตามลักษณะของการใช้งาน ซึ่งผลของการทดสอบนี้จะให้ throughput ต่าง ๆ ออกมา อันเนื่องมาจากการทดลองเปลี่ยนส่วนประกอบของเน็ตเวอร์ก เช่น แคช, ไดรฟ์, เน็ตเวอร์กการ์ด, เวอร์กสเตชั่น, เซอร์ฟเวอร์โปรเซสเซอร์ และอื่น ๆ เพื่อจะได้สามารถใช้ผลของการทดลอง หรือนำ adjustments เหล่านี้ไปปรับปรุงประสิทธิภาพของ LAN ต่อไป

ผลที่ได้เมื่อวัดด้วย LANPERF

ฮาร์ดแวร์ที่ใช้ในการทดสอบจะเป็นเวอร์กสเตชั่นแบบ 8088, 80286 80386 บน Token-Ring และใช้ Compaq Deskpro แบบ 386/16 เป็นไฟล์เซอร์ฟเวอร์ พร้อมด้วยแรมขนาด 6 เมกะไบต์ โอเปอเรติ้งซิสเต็มของเน็ตแวร์ 2.1 บนเครื่อง 386/16 นั้นจัดให้เป็นไปตามมาตรฐานที่วางไว้แล้ว ซึ่งถือเป็นปกติธรรมดาสำหรับกรณีนี้ เช่น มี routing

buffers เท่ากับ 80, สามารถเปิดไฟล์พร้อม ๆ กันได้ถึง 300 ไฟล์ เป็นต้น

การทดสอบระลอกแรกเป็นการวัด throughputs ของสเตชั่นเดี่ยวในเน็ตเวอร์กที่ไม่มีโหลดเน็ตเวอร์กอื่น ๆ เข้ามาเกี่ยวข้อง โดยโปรแกรมชื่อ LANPERF จะเป็นผู้กำหนดหรือสร้างเส้นทางการจราจรขึ้นมาเอง และจะมีการวัด throughput เป็นพันไบต์ต่อวินาที (KB/s)

การทดสอบตอนแรกนี้เป็นการหา base line readings เพื่อให้ได้ข้อมูลขั้นพื้นฐานต่าง ๆ จึงต้องใช้เครื่องที่เป็นฮาร์ดไดรฟ์แบบ 10 เมกะเฮิรตซ์ AST Premium/286 โดยยังไม่ได้โหลดส่วนที่เรียกว่าเน็ตเวอร์กเชลล์เข้าไปในเครื่อง และในกรณีที่ขนาดของบล็อก (block size) เท่ากับขนาดไฟล์นี้ throughput ของการทดสอบในแง่ของการอ่าน (read test) ที่ local drive จะเป็นดังรูปที่ 1 throughput สำหรับ block size ที่มีขนาดต่ำกว่า 512 ไบต์ลงไปที่ดูค่อนข้างสูงนั้นเกิดจาก DOS buffering (อันเป็นเทคนิคในด้านโปรแกรมมิ่งที่ใช้แก้ปัญหาความล่าช้าในการอ่านหรือเขียนข้อมูลสำหรับเพอริเฟอรัลดีไวซ์ต่าง ๆ) ส่วน throughput สำหรับ write test นั้นจะต่ำกว่า read test ราว 5-10% ซึ่งก็ยังคงทดสอบโดยอาศัยประโยชน์จากเทคนิคที่เรียกว่า local buffering เหมือน read test ในตอนแรก

ต่อไปก็เป็นการทดสอบ local drive โดยโหลดตัว NetWare Shell เข้าไปในเครื่อง ตอนนี้เองที่ประสิทธิภาพเริ่มลดลงขณะ buffered operations อันเนื่องมาจากการที่ shell ของเน็ตแวร์ไปดักจับ DOS operations ต่าง ๆ ไว้ ปรากฏการณ์นี้เกิดขึ้นทั้งไดรฟ์ที่เป็น local และในเน็ตเวอร์ก สรุปแล้วเมื่อโหลดส่วนที่เป็น shell เข้าเครื่องในรูปของ buffered operations ที่ local drive ก็จะทำให้ช้าลงราว 10-20%

throughput ของ read test บนสเตชั่นเดิมในกรณีของเน็ตเวอร์กไดรฟ์จะเป็นดังรูปที่ 2 จะเห็นว่าในช่วงของ buffered operations นั้น network throughput จะสูงกว่าแบบ local แต่เมื่อ block size เพิ่มขึ้น throughput ก็ไม่ได้เพิ่มเป็นเงาตามตัวหรือในสัดส่วน

BLOCK SIZE (bytes)	THROUGHPUT (KB/s)
1	1.37
10	13.56
64	84.12
256	302.74
512	478.30
1,024	59.70
2,048	119.50
4,096	119.60

รูปที่ 1 throughput สำหรับ read test แบบ local drive

BLOCK SIZE (bytes)	THROUGHPUT (KB/s)
1	2.73
10	26.84
64	162.26
256	534.78
512	64.20
1,024	97.40
2,048	98.50
4,096	96.60

รูปที่ 2 throughput ของ read test แบบ network drive

ที่ควรจะเป็น เนื่องจากต้องผ่านเซอร์ฟเวอร์ของเน็ตเวอร์กอันมีส่วนทำให้ช้าลง

buffered blocks ของ NetWare Shell จะมีขนาดเท่ากับ 256 ไบต์ หรือน้อยกว่านี้ เมื่อเป็นเช่นนี้ overlaid operations ต่าง ๆ ที่ขนาดของ block ยาวตั้งแต่ 257 ไบต์ขึ้นไปก็จะต้องไปที่เซอร์ฟเวอร์ก่อน อันเป็นการเสียเวลาอย่างหนึ่ง

buffering ของ shell ในเน็ตแวร์ 2.1 นั้นได้ optimize ไว้สำหรับ file operations เพื่อการใช้งานเน็ตเวอร์กเพียงบางแบบเท่านั้น การอ่านและเขียนแบบ sequencial ชนิดที่มีความยาวมาก ๆ นั้นจะให้ throughput ที่ไม่มากไปกว่าในกรณีของ overlaid operations ซึ่งในช่วง overlaid นี้จะไม่มีปัญหาในเรื่องของ buffering แต่อย่างใด

สรุปแล้วสิ่งที่จะกำหนด throughput สำหรับ operations ต่าง ๆ ดังได้กล่าวมาเป็นตัวอย่างในข้างต้น และ operations อื่น ๆ จะขึ้นกับเซอร์ฟเวอร์, เวอร์กสเตชัน และเน็ตเวอร์กลิงก์เป็นสำคัญ อันเป็นการบอกว่าการใช้เน็ตเวอร์กนั้นจะต้องพึ่งฮาร์ดแวร์ด้วย เช่น ถ้าใช้สเตชั่นแบบ 10 เมกะเฮิรตซ์ Premium/286 ก็จะให้ throughput ราว 100 KB/s ในการทดสอบที่กล่าวมาแล้วในข้างต้น แต่พอลองไปใช้ Compaq Deskpro 386/16 มาวิ่งงานแบบเดียวกับข้างต้นก็จะได้ throughput ออกมา 128.2 KB/s เป็นต้น

ผลการทดสอบแบบหลาย stations

การทดสอบสมรรถนะของเน็ตแวร์ 2.1 ในขั้นต่อไปเป็นการกระทำกับ configurations ที่มีหลาย ๆ สเตชั่น โปรแกรม LANPERF จะประสานงานให้การทดสอบเป็นไปในรูปที่ว่าสเตชั่นทุกอันจะเริ่มและหยุดพร้อมกันหมด โดยได้ทำการทดสอบถึง 12 สเตชั่นที่นำมาพ่วง

เข้าไว้ด้วยกันในเน็ตเวอร์กอันหนึ่ง ซึ่งประกอบด้วยฮาร์ดแวร์ เช่น Compaq Deskpro 386/16, 8 เมกะเฮิรตซ์ IBM PC/AT, IBM PS/2 Model 50, 10 เมกะเฮิรตซ์ AST Premium/286 และ IBM PC 4.77 เมกะเฮิรตซ์

ตัวเลข throughput ที่ได้นั้น สามารถแสดงผลกระทบอันเนื่องมาจากการที่สเตชั่นเป็นจำนวนมาก (12 เครื่อง) ต้องอาศัยสื่อในเน็ตเวอร์กเพื่อรับส่งข้อมูล และตัวเซอร์ฟเวอร์ในลักษณะของการใช้งานร่วมกัน ผลกระทบอันนี้จะแสดงออกมาในรูปของความล่าช้า เช่น เครื่องแบบ Premium/286 ที่เคยให้ throughput 100 KB/s เวลามีสเตชั่นเดียวก็จะให้ throughput ลดลงเหลือแค่ 20.4 KB/s เมื่อทดสอบท่ามกลางเวอร์กสเตชั่นเป็นจำนวน 12 สถานีด้วยกัน

การทดสอบ multistation นั้นบอกสิ่งที่น่าสนใจและสำคัญต่อการใช้งานจริงได้หลาย ๆ อย่าง ซึ่งอย่างน้อยก็บอกได้ในขั้นต้นว่า ความเร็วของเครื่อง หรือชนิดของโปรเซสเซอร์จะไม่มีผลในการเพิ่ม throughput แต่อย่างใด เช่น จากการทดสอบแบบ 10-station ที่ประกอบด้วยเครื่องแบบ 8 เมกะเฮิรตซ์ 80286 และแบบ 16 เมกะเฮิรตซ์ 80386 ก็จะให้ throughput เฉลี่ยออกมาเป็น 24.3 KB/s และให้การเบี่ยงเบนเฉลี่ยจากค่าเฉลี่ยดังกล่าวออกมาเท่ากับ 0.1 KB/s แสดงว่าสเตชั่นแบบ 80286 หรือ 80386 แบบใดแบบหนึ่งอาจให้ throughput อย่างสม่ำเสมอภายใต้ภาระการใช้งานอันหนักข้อยกเว้นในงานนี้ก็มีอยู่เหมือนกันคือเครื่องพีซีแบบ 4.77 เมกะเฮิรตซ์จะช้ากว่าเครื่องแบบ 16 บิตและ 32 บิตราว 10 ถึง 20% ในการทดสอบการใช้งานทุกแบบ

สรุปแล้วในการทดสอบแบบสถานีเดี่ยวนั้น throughput จะขึ้นกับความเร็วของโปรเซสเซอร์ ส่วนในกรณีของการทดสอบสถานีจำนวนมาก ๆ ตัวการที่จะมาจำกัด throughput จะขึ้นอยู่กับเซอร์ฟเวอร์หรือการเชื่อมโยงข้อมูล ไม่ใช่ที่สถานีย่อย

การทดสอบปลีกย่อยอื่น

การทดสอบขั้นต่อไปเป็นการหา aggregate server throughput ซึ่งทำได้โดยรวม throughput ของสเตชั่นต่าง ๆ ในเน็ตเวอร์กเข้าด้วยกันหมายถึงการทดสอบแบบ group test นั่นเอง ส่วนการทดสอบที่เป็น read test สำหรับ block size ใหญ่ ๆ (1KB, 2KB, 4KB) สำหรับกรณีนี้จะให้ aggregate server throughput ที่แปรในเชิงเส้นตรงตั้งแต่ 240 ถึง 245 KB/s การแปรจำนวนสเตชั่นในระดับที่เหนือจากจุดต่ำสุดของสเตชั่นจะมีผลกับ throughput ของแต่ละสเตชั่น และไม่มีผลกับตัวเลขของ throughput ของตัวเซอร์ฟเวอร์ที่เป็น aggregate แต่อย่างใด (aggregate server throughput)

ในขณะที่ทำการทดสอบแบบ multiple-station พบว่า throughput ของเครื่องแต่ละเครื่องนั้นสามารถหาได้จากการเอาจำนวนสเตชั่นไปหารตัวเลข 240 เช่น การทดสอบแบบ 12 สเตชั่นจะให้ throughput ออกมาราว 20 KB/s และการทดสอบแบบ 6 สเตชั่นจะให้ throughput ออกมา 40 KB/s เป็นต้น นั่นคือหากมีจำนวนสเตชั่นยิ่งมาก throughput ยิ่งตก

นอกจากนี้การทำ write test ในกรณีของหลาย ๆ สเตชั่นนั้นจะไม่มีการอิ่มตัวในเน็ตเวอร์กแต่อย่างใด เมื่อเทียบกับการ read test อันอาจเนื่องมาจาก disk latency ซึ่งเป็นสาเหตุของความล่าช้า aggregate server throughput สูงสุดของการเขียนที่ได้จะตกราว 210 KB/s

ต่อมาลองดูที่ผลลัพธ์ ประเภทของเน็ตเวอร์กหรือ topology ที่จะมีอิทธิพลต่อ throughput โดยในขณะทดสอบนั้นจะตั้งสมมติฐานว่าการใช้ console ที่ตัวเซอร์ฟเวอร์ต้องไม่เกินร้อยละ 75 ในกรณีที่ใช้ topology แบบ Token-ring ทำให้สามารถประมาณได้ว่า throughput ของเซอร์ฟเวอร์แบบ aggregate ที่มีค่าเท่ากับ 240 KB/s นี้แสดงถึงการอิ่มตัว หรือขีดจำกัดอันเนื่องมาจาก test topology แต่ไม่ได้เป็นเหตุอันเนื่องมาจากเซอร์ฟเวอร์แต่อย่างใด

ในแง่ของ topology นี้ Novell เคยทำรายงานว่าได้ทำการทดสอบ LAN เหมือนกับ PC Tech Journal เช่นกัน โดยใช้เซอร์ฟเวอร์แบบ 80386 ที่มีงานโหลดอยู่เต็มที่บนเน็ตเวอร์กแบบ Token-ring และได้ aggregate throughput ที่สูงถึง 270 KB/s

อิทธิพลของ topology

เพื่อให้ผู้อ่านมองเห็นอย่างเด่นชัดว่า topology หรือ network architecture นั้นมีอิทธิพลอย่างมากกับ throughput ของเน็ตเวอร์ก Novell ได้ทำการทดสอบเน็ตแวร์ของตนกับเน็ตเวอร์กประเภท Ethernet โดยใช้กลุ่มเซอร์ฟเวอร์ที่เป็นเครื่องแบบ 80386 และก็ได้ throughput ออกมาถึง 660 KB/s หรืออาจมากกว่านี้ ซึ่งแสดงว่าบอร์ด Token-ring และไดรเวอร์ของไอบีเอ็มนั้นอิ่มตัว

ก่อนที่เซอร์ฟเวอร์แบบ 80386 จะมีโอกาสให้ throughput ออกมาได้เต็มที่

เนื่องจาก 6 เมกะเฮิร์ตซ์ 80286 NetWare ตัวเซอร์ฟเวอร์บน Token-ring นั้นได้ก้าวไปถึงการใช้งานเต็ม 100% ในขณะเดียวกับที่เซอร์ฟเวอร์แบบ 80386 ที่ใช้ NetWare เป็น OS ยังสามารถให้ throughput ต่อไปอีกเรื่อย ๆ โดยที่บอร์ด Token-ring และไดรเวอร์ของไอบีเอ็มนั้นทำอะไรต่อไปอีกไม่ได้แล้ว สรรพคุณของ IBM Token-ring ที่มาใช้กับ NetWare จึงยังจัดว่าทำงานไม่ได้เร็วที่สุด หรือเร็วเท่าที่ควรแม้จะไม่ช้าก็ตาม

สรุปผลการทดสอบเน็ตแวร์ 2.1

จากการทดสอบเกือบทุกแง่มุมโดยหน่วยทดสอบของ PC Tech Journal นั้นโดยทั่วไปพบว่า NetWare ทำงานได้ดีมากทั้งในกรณีสเตชั่นเดียวและหลายสเตชั่น คือการทดสอบแบบ single-station นั้นให้ผลสรุปได้ว่า throughput ที่ให้ออกมานั้นจะพอ ๆ กับ throughput ของ DOS operations บน local drive ชนิดเร็ว

ส่วนการทดสอบแบบ multistation นั้นเป็นการแบ่งปันทรัพยากรต่าง ๆ เช่น ไฟล์, พรินเตอร์ ฯลฯ ของเซอร์ฟเวอร์ให้กระจายไปยังสเตชั่นต่าง ๆ อย่างเท่า ๆ กันท่ามกลาง configurations หรือส่วนประกอบในด้านฮาร์ดแวร์ภายในเน็ตเวอร์กที่ต่างกัน ดังได้แสดงผลของการทดสอบที่เป็นตัวเลข พร้อมด้วยการให้อรรถาธิบายไว้แล้วข้างต้น

ดังนั้นจึงสรุปได้ว่าประสิทธิผลของการใช้งานของ NetWare นั้นก็เหมือนกับเน็ตเวอร์กซอฟต์แวร์อื่น ๆ ที่มักจะได้รับผลกระทบจากแฟกเตอร์ต่าง ๆ เช่น local caching, network load และ topology ซึ่งเป็นข้อเตือนใจแก่ผู้พัฒนาซอฟต์แวร์ที่จะมาใช้ในเน็ตเวอร์ก (network application software) ให้คำนึงถึงแฟกเตอร์ทั้งสามให้มากเพราะจะเกี่ยวข้องกับเรื่องของ optimizing performance อย่างเลี่ยงไม่ได้

ขีดความสามารถเน็ตแวร์ในด้านต่าง ๆ

ผู้เขียนได้กล่าวถึงการนำแนวความคิดใหม่ ๆ ในแง่ของการทำเบนช์มาร์คโดยอาศัย performance utility ที่เรียกว่า LANPERF (LAN Performance) เพื่อวัดประสิทธิผลการใช้งานของ NetWare ในทุกแง่ทุกมุมแล้ว หลังจากนี้จะขอกล่าวถึงขีดความสามารถของเน็ตแวร์ในส่วนที่ยังไม่ได้พูดถึง เช่น ความสามารถของซอฟต์แวร์อันนี้ในด้านการเชื่อมโยงเน็ตเวอร์กต่าง ๆ เข้าด้วยกัน (internetworking capability), protocol layers, NetWare internals และเรื่องของโปรแกรมมิ่งภายใต้เน็ตแวร์

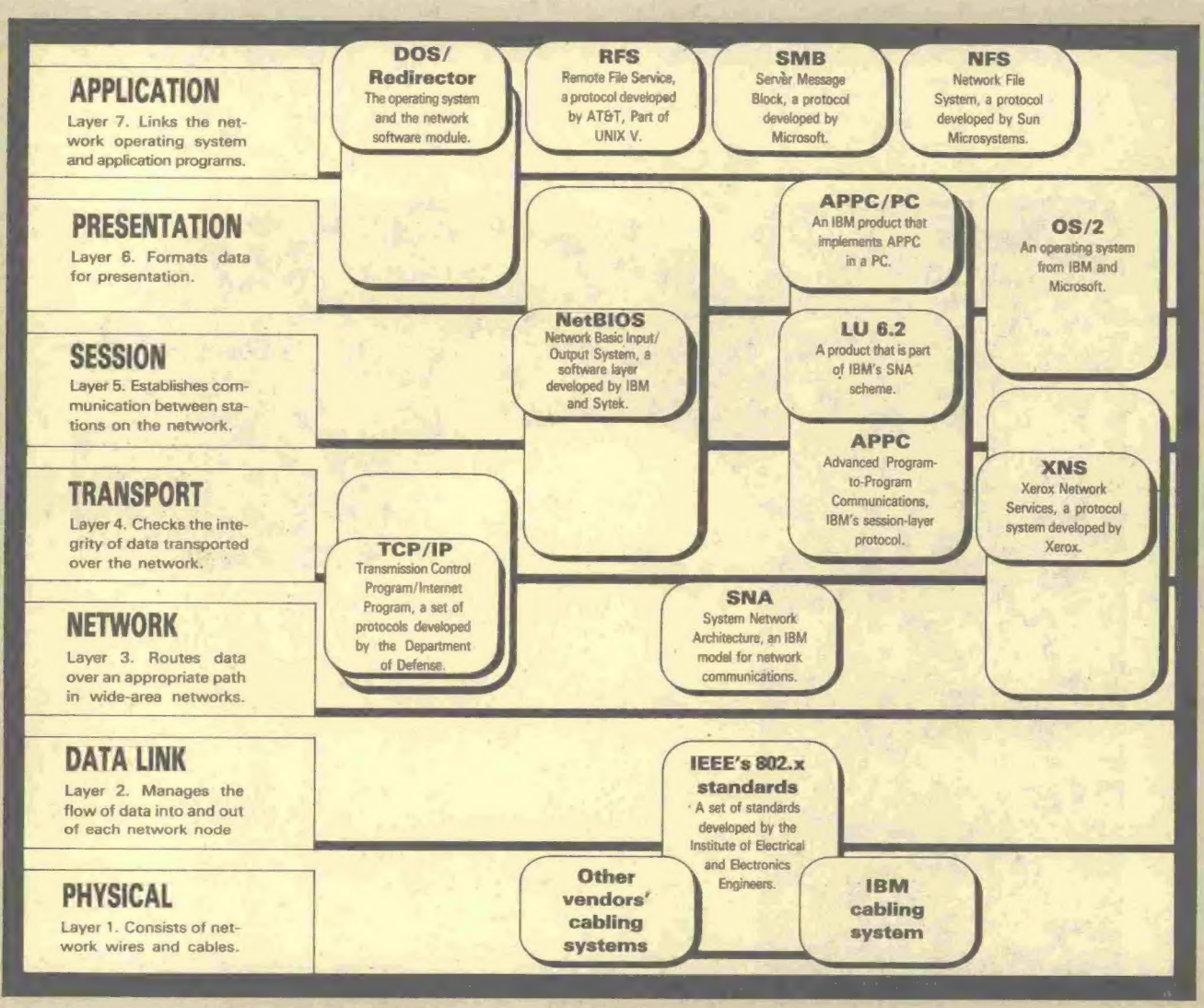
เน็ตเวอร์กท่ามกลางเน็ตเวอร์ก

NetWare สามารถสนับสนุนการเชื่อมโยงระหว่าง LAN ต่าง ๆ หรือหลาย ๆ เน็ตเวอร์ก โดยอาศัยบริดจ์ (bridges) หรือสะพานที่อยู่ในไฟล์เซอร์ฟเวอร์ หรือติดตั้งอยู่ในพีซีที่ทำหน้าที่อันใดอันหนึ่งโดยเฉพาะ ถือเป็น internetworking capability อย่างหนึ่งซึ่งความสามารถเช่นนี้จะแผ่ไปถึงการบริการในแง่ของไฟล์และงานพิมพ์ต่าง ๆ ตลอดจนงานด้านบริหารของเน็ตเวอร์กด้วย

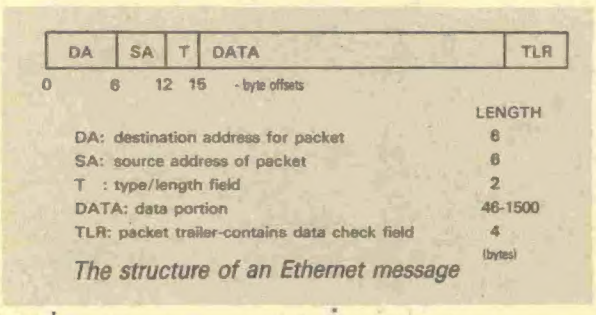
บริดจ์ของเน็ตแวร์นั้นจะทำหน้าที่เหมือน router ในทัศนะของ Open System Interconnection (OSI) โดยจะทำหน้าที่ในระดับเน็ตเวอร์กอันเป็นระดับที่ 3 (layer 3) ของ OSI โมเดล ดังแสดงในรูปที่ 3 ซึ่งจุดมุ่งหมายของโมเดลอันนี้ก็คือการกำหนดมาตรฐานกลุ่มหนึ่งขึ้นมาเพื่อให้คอมพิวเตอร์ที่ตั้งอยู่ตามจุดต่าง ๆ ในเน็ตเวอร์กสามารถพูดจาสื่อความกันได้ ดังนั้นตามความหมายของ OSI โมเดลแสดงว่าบริดจ์ที่ใช้ในเรื่องของเน็ตแวร์จะทำหน้าที่ส่งข้อมูลต่าง ๆ ให้ผ่านไปในเส้นทางที่เหมาะสมในเน็ตเวอร์กคือเป็น router อย่างหนึ่งนั่นเอง

การทำงานของบริดจ์ในเน็ตแวร์จะมีซอฟต์แวร์ของเน็ตแวร์เป็นตัวควบคุม ซึ่งเรียกว่า router software ที่รู้แอดเดรสของ routers ต่าง ๆ ในแต่ละเน็ตเวอร์ก (ขณะนี้เรากำลังพูดถึงกรณีที่มีหลาย ๆ เน็ตเวอร์กเชื่อมโยงกันอยู่) โดยจะทำหน้าที่สกัดกั้นกลุ่มของสัญญาณ (packets) ที่ไม่มีหน้าที่ต้องผ่านเข้ามาที่บริดจ์อันใดอันหนึ่งด้วยการดูจาก address header ของแต่ละกลุ่มสัญญาณ ซึ่งผู้อ่านสามารถดูตัวอย่างจากรูปที่ 4 ประกอบ เพื่อให้นึกภาพออกว่าโครงสร้างของการส่งสัญญาณในรูปของกลุ่มสัญญาณนั้นมีอะไรบ้าง จะเห็นว่าสิ่งที่เรียกว่า address header ตามภาษาของเน็ตแวร์นั้นก็คือส่วนที่เป็น DA (destination address for packet) ในรูปที่ 4 นั่นเอง ซึ่งจะบอกจุดหมายปลายทางของกลุ่มสัญญาณว่าจะไปที่ไหน

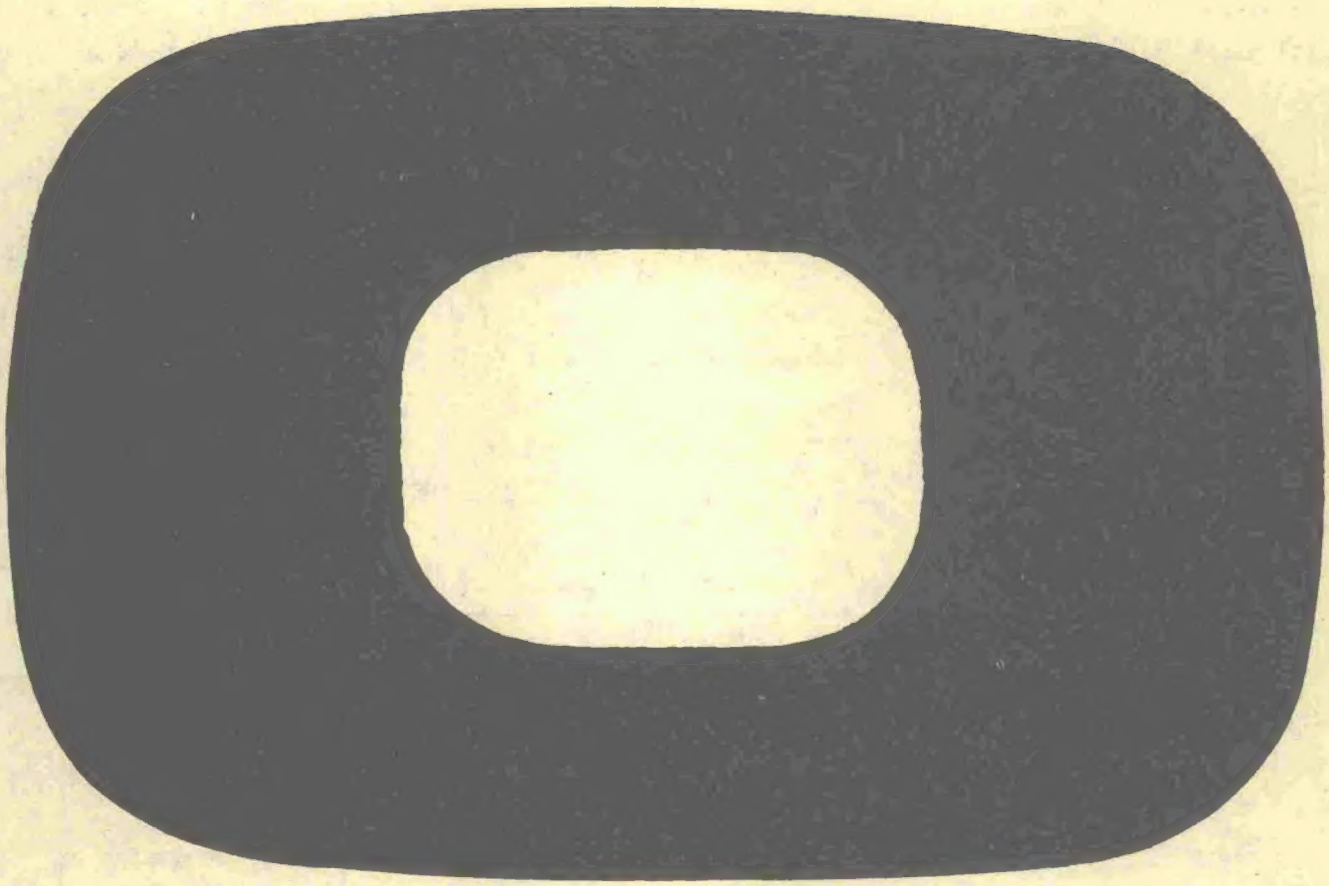
บริดจ์อันใดอันหนึ่งในเน็ตแวร์จะสามารถเชื่อมโยงเน็ตเวอร์กทั้งที่เหมือนกันและต่างกันได้ไม่เกิน 4 เน็ตเวอร์ก เช่น สามารถเชื่อมโยงเซอร์ฟเวอร์แต่ละ



รูปที่ 3 แผนผังแสดง OSI โมเดลสำหรับ Network Communications



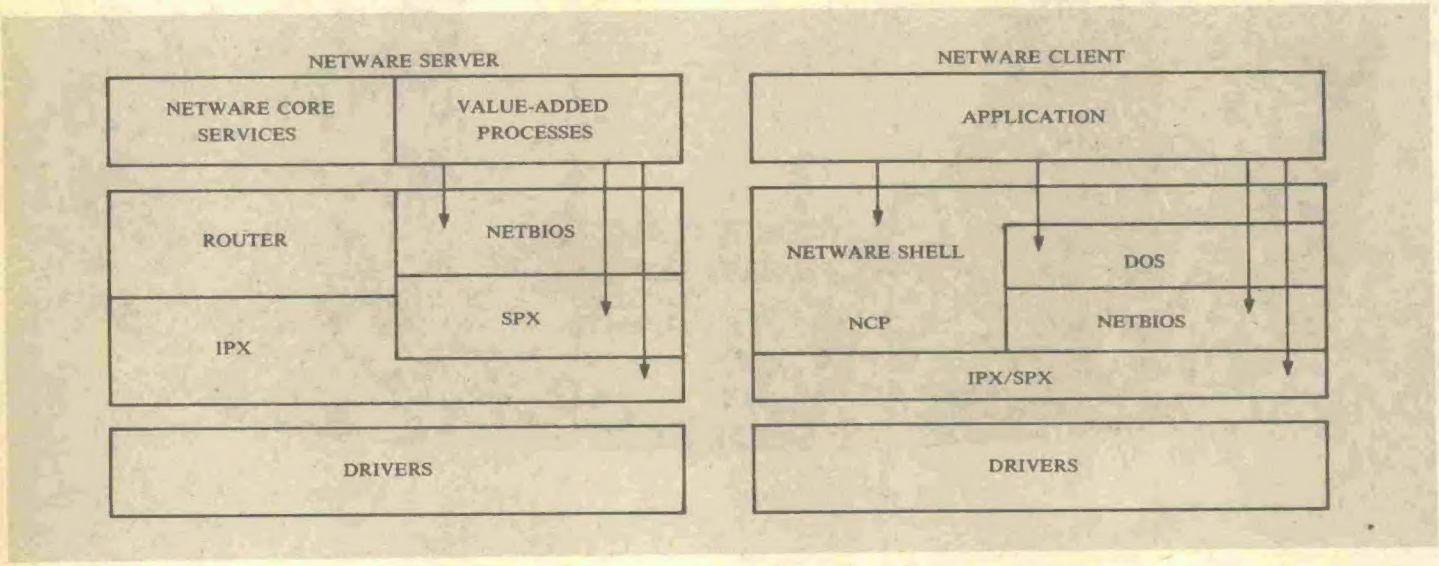
รูปที่ 4 ตัวอย่างของการส่งข้อมูลในรูปของ packet ซึ่งขอยืมหลักการมาจากที่ใช้อยู่ใน LAN แบบหนึ่งที่ชื่อว่า Ethernet



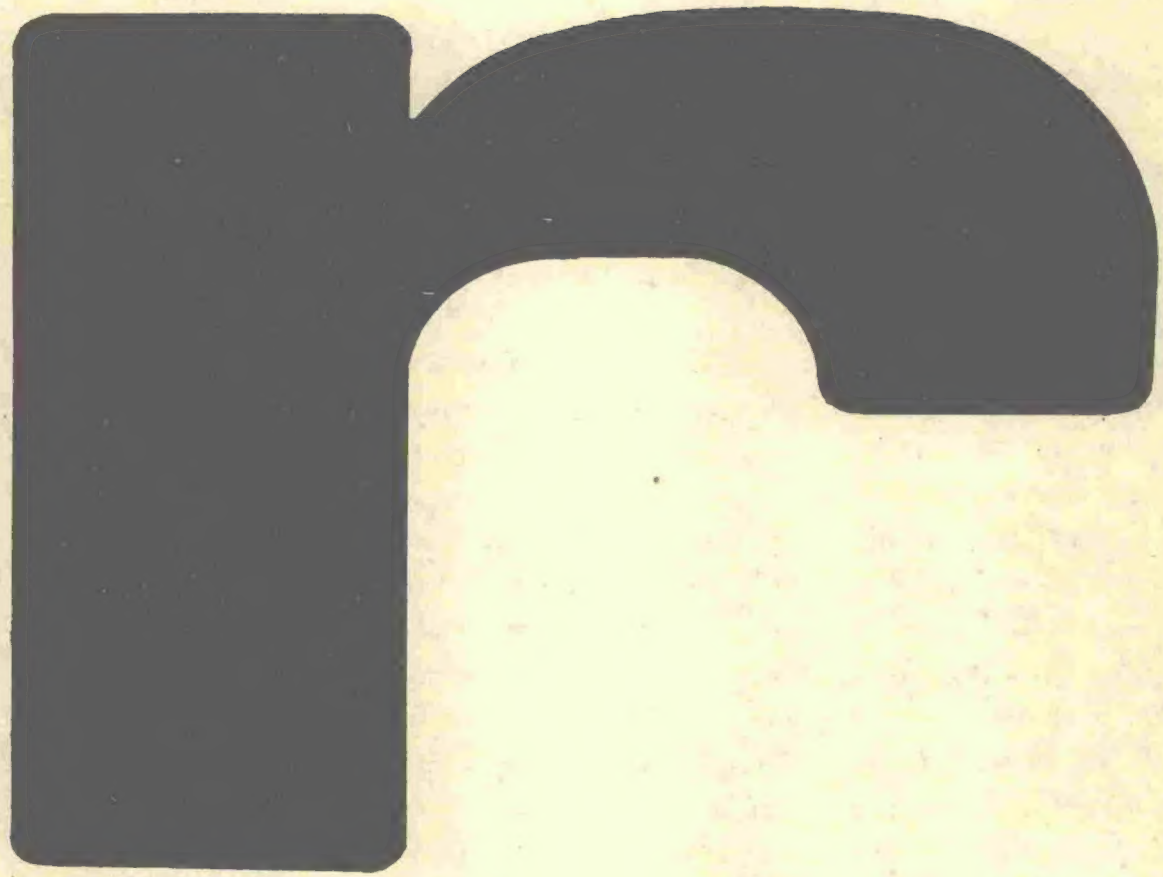
จุดของ ARCnet ให้เข้ากับ backbone ของ Ethernet ที่ทำงานด้วยความเร็วสูง ในกรณีเช่นนี้ router ซอฟต์แวร์ที่อยู่ในเซอร์ฟเวอร์ต่าง ๆ จะส่งกลุ่มสัญญาณไปยัง backbone เฉพาะที่จำเป็นเท่านั้น

นอกจากนี้เน็ตแวร์ยังมีความสามารถค่อนข้างมากในการเชื่อมโยง topologies หรือรูปแบบของเน็ตเวอร์กที่ต่าง ๆ กันออกไป อนึ่ง ส่วนของซอฟต์แวร์ภายในตัวเน็ตแวร์เพียงส่วนเดียวที่จะต้องเปลี่ยนตาม topologies ก็คือตัวไดรเวอร์ ดังได้แสดงในรูปที่ 5 ส่วน protocol อื่น ๆ ที่อยู่เหนือขึ้นไปจากไดรเวอร์จะไม่เปลี่ยนตาม topologies จุดนี้เองที่เริ่มมีปัญหาที่จะทำให้การเชื่อมโยงเน็ตเวอร์กที่มี topologies ต่าง ๆ กันนั้นยากขึ้น คือยากในแง่ของการวินิจฉัยสาเหตุขัดข้องต่าง ๆ และยากในการซ่อมบำรุงขณะใช้งาน เคยมีผู้ให้ความเห็นว่า การเชื่อมโยงด้วยบริดจ์ตามแบบของเน็ตแวร์เท่าที่มีอยู่ในขณะนี้ (ปี 1988) น่าจะประหยัดกว่าการเชื่อมโยงสเตชั่นต่าง ๆ ด้วย high bandwidth links หรือแบบอื่น

ปกติแล้วบริดจ์ของเน็ตแวร์หรือเซอร์ฟเวอร์ตัวหนึ่งจะมีเน็ตเวอร์กอะแดปเตอร์ได้ 4 ตัว แต่การนำบอร์ดอันเป็นผลิตภัณฑ์แบบต่าง ๆ มาผสมเพื่อใช้งานร่วมกันนั้นจะบังคับให้ต้องลดจำนวนเช่นนี้ลงไปพอสมควร เช่น จะสามารถนำบอร์ดแบบ 3 Com's 8-bit 3C505 ได้แค่ 3 ตัวมาต่อเข้ากับบริดจ์หรือเซอร์ฟเวอร์ หรือถ้าจะนำบอร์ดของไอบีเอ็ม Token-ring มาใช้ในสภาพของเน็ตแวร์ก็จะทำได้แค่สองบอร์ดเท่านั้น แต่ถ้าจะสร้าง configuration ให้มีมากได้ถึง 4 บอร์ดก็ต้องใช้บอร์ดเหล่านี้ เช่น ใช้การ์ดของ Standard Micro System ARCnet 2 อัน ใช้บอร์ดของ Token-ring หนึ่งอัน และใช้บอร์ดของ 3Com 3C505 หนึ่งอัน ซึ่งเซอร์ฟเวอร์ที่จะมาสนับสนุนการใช้บอร์ด 4 อันพร้อม ๆ กันนี้จะสามารถรับใช้เน็ตเวอร์กได้ 4 เน็ตเวอร์ก สองเน็ตเวอร์กแรกจะเป็น ARCnet ส่วนเน็ตเวอร์กอันที่สามจะ



รูปที่ 5 NetWare Block Diagram ซึ่งแสดง protocol stack ของ NetWare ทั่ว ๆ ไป โดยแสดงสภาพของเน็ตแวร์ที่อยู่ในตัวเซอร์ฟเวอร์และเน็ตแวร์ที่อยู่ใน client (หรือในสเตชั่นต่าง ๆ ที่ผู้ใช้จะเข้าไปใช้ได้) ลูกศรที่ปรากฏอยู่ในรูปนั้นเกิดจากการที่แอพพลิเคชั่นต่าง ๆ เรียกใช้ facilities ที่มีอยู่ เช่น ในด้านผู้ใช้ที่เป็น client side นั้น แอพพลิเคชั่นจะเรียกไปที่ DOS, NETBIOS, IPX/SPX หรือ NetWare Shell ส่วนเซอร์ฟเวอร์สแต็กนั้นมีหน้าที่สนับสนุน core services ต่าง ๆ โดยอาศัย value-added process ของผู้ผลิตซอฟต์แวร์ซึ่งเป็นบุคคลที่ 3 ที่จะมาใช้วิ่งอยู่ในตัวเซอร์ฟเวอร์



เป็น Token-ring และอันสุดท้ายจะเป็น Ethernet

ถ้าเป็นไปได้บริดจ์ของ NetWare ควรประกอบด้วย dedicated PC หนึ่งตัว ในกรณีที่ไม่สามารถใช้ dedicated PC ตัวหนึ่งให้มาทำหน้าที่นี้โดยเฉพาะก็ยังมีทางเลี่ยงได้ด้วยการใช้ non-dedicated PC ซึ่งเน็ตแวร์ก็ให้การสนับสนุนเช่นกัน ในกรณีนี้ต้องสำรองเมมโมรีไว้ 160 KB สำหรับเก็บ bridge software อันจะมีปัญหาตามมาคือ ไม่สามารถสนับสนุน remote sessions ได้อีกต่อไป และจะเกิดปัญหา data loss อีกด้วย ซึ่งผู้อ่านจะหาความกระจ่างในเรื่องของ session ได้จากรูปที่ 3

เนื่องจากการกล่าวถึงหัวข้อต่อ ๆ ไป เช่น เรื่องของ protocol layers นั้น จะเกี่ยวกับเลเยอร์ต่าง ๆ ดังแสดงไว้ในรูปที่ 3 เป็นอย่างมาก ซึ่งแต่ละเลเยอร์ดังในรูปที่ 3 นั้นเป็นการแสดงหน้าที่ในระดับต่าง ๆ ตั้งแต่ physical จนถึงแอพพลิเคชั่นว่ามีอะไรบ้าง และในแต่ละระดับนับตั้งแต่ physical layer ขึ้นไปนั้นได้มีผลิตภัณฑ์ทั้งฮาร์ดแวร์และซอฟต์แวร์อะไรมาให้การสนับสนุนบ้าง การที่ต้องอิงถึงรูปที่ 3 อยู่บ่อย ๆ เวลาพูดถึงเรื่องของ LAN หรือเรื่องของเน็ตแวร์ ก็เพราะต้องการยึดมาตรฐานอันใดอันหนึ่งเพื่อให้พูดกันรู้เรื่อง และต้องเป็นมาตรฐานที่คนส่วนใหญ่ยอมรับ ซึ่งในกรณีนี้จะเป็นมาตรฐานของเน็ตเวอร์กคอมมูนิเคชั่นที่เกิดจากการบัญญัติของ International Standards Organization (ISO) รูปที่ 3 จึงเกี่ยวข้องผูกพันอย่างมากเมื่อเรากล่าวถึงเรื่องต่าง ๆ ในเน็ตแวร์

Protocol Layers

แนวความคิดในเรื่องของ protocol layers ของ NetWare (รูปที่ 5) ก็คล้าย ๆ กับการแบ่งเลเยอร์ต่าง ๆ ของ OSI โมเดล ดังรูปที่ 3 ซึ่งแตกต่างจากรูปที่ 5 ที่ดูเข้าใจยากเพราะมีหลายสัดหลายส่วน แต่ถ้าเราจะหันมาสนใจว่าแต่ละสัดส่วนของรูปที่ 5 นั้นมีหน้าที่หลักสำคัญอะไรบ้าง เห็นจะเข้าใจอะไรได้ไม่ยากนัก โดยจะเริ่มจากสิ่งที่เรียกว่า IPX/SPX เสียก่อน แล้วจึงพูดถึงส่วนอื่น ๆ ภายในบล็อกไดอะแกรมของเน็ตแวร์ที่แสดงในรูปที่ 5 ต่อไปภายหลัง

IPX/SPX เป็น subnetwork ซอฟต์แวร์ภายในเน็ตแวร์ที่ย่อมาจากคำว่า internet packet exchange/sequenced packet exchange ซึ่งตอนนี้ผู้อ่านควรดูรูปที่ 3 และ 5 ประกอบไปเรื่อย ๆ เพื่อสร้างความเข้าใจ เพราะเกี่ยวข้องกันมาก และทุกครั้งที่พูดถึง OSI โมเดลก็หมายถึงรูปที่ 3

IPX ของเน็ตแวร์จะสัมพันธ์กับเน็ตเวอร์กเลเยอร์ของ OSI โมเดล คือมีหน้าที่สนับสนุน internetwork routing และ nonguaranteed packet delivery ส่วน SPX นั้นสัมพันธ์กับ transport layer ของ OSI โมเดลและทำหน้าที่สนับสนุน IPX ด้วยการทำให้เกิด packet sequencing และ guaranteed packet delivery ขึ้นมา ทั้ง IPX และ SPX ต่างก็เป็น protocol layer ทั้งคู่โดยพัฒนามาจาก Xerox Network Systems (XNS) อันเป็นข้อกำหนดในด้าน internetwork protocol ที่ Xerox ใช้อยู่ XNS นี้ปรากฏอยู่ใน transport layer ของรูปที่ 3 เช่นกัน

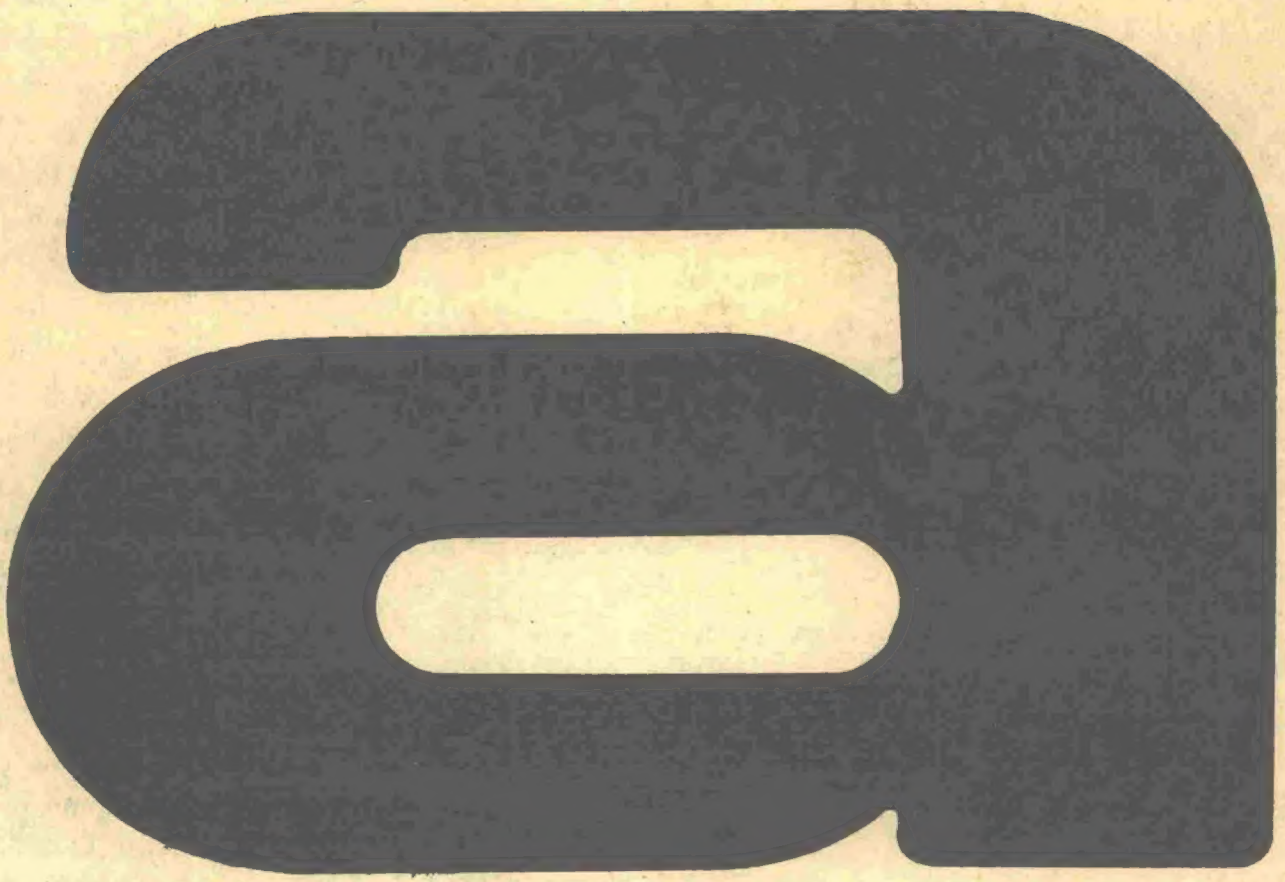
ต่อไปก็เป็นเรื่องของ NetWare shell ซึ่งหน้าที่ของ shell ก็คือการเข้ามาช่วยเพิ่มให้ประสิทธิผลในการใช้งานสูงขึ้น และช่วยให้การใช้งานในด้านต่าง ๆ มีมากขึ้น shell อันนี้จะใช้เฉพาะแต่ตัว IPX สำหรับติดต่อกับไฟล์เซอร์ฟเวอร์ ทั้งนี้เนื่องจากกระบวนการทำงานในตัว shell นั้นจะมีรูทีนที่เข้ามาทำหน้าที่ในด้าน internal retry และ error-recovery อยู่แล้ว ส่วนตัว SPX ที่ shell ไม่ได้นำมาใช้นี้จะถูกโหลดเข้าไปในเวอร์กสเตชั่น เพื่อสนับสนุนการใช้งานที่เป็นแอพพลิเคชั่นต่าง ๆ สำหรับผู้ใช้แต่ละราย ซึ่งจะต้องอินเตอร์เฟสกับทรานสปอร์ตเลเยอร์

การใช้งานที่เรียกว่า client applications ที่กล่าวในข้างต้นนี้ ต้องการการสนับสนุนจาก SPX ในแง่ของ guaranteed delivery และ flow control ในกรณีที่มีการติดต่อกับ nodes ต่าง ๆ ที่ใช้เวลาในการตอบรับ (response time) ค่อนข้างนาน หรือเป็นไปอย่างไม่แน่นอน เช่น communication gateways ต่าง ๆ, ฐานข้อมูล, optical-disk หรือ archive servers

สิ่งที่จะมาทำหน้าที่เป็นอีมูเลเตอร์สำหรับผลิตภัณฑ์ของ Novell ก็คือ NETBIOS ซึ่งจะรับภาระในด้านการสื่อสารระหว่างโปรแกรม (interprogram communications) อันเป็นโปรโตคอลที่อยู่เหนือขึ้นไปจาก IPX อีกทีหนึ่ง แอพพลิเคชั่นต่าง ๆ จะใช้ NETBIOS สำหรับการสื่อสารที่เป็น interprocess กับ nodes อื่น ๆ ในเน็ตแวร์เท่านั้น

NetWare internals

เหนือขึ้นไปจาก communications layers ที่เป็น IPX/SPX นั้นจะเป็นที่อยู่ของ NetWare Shell หน้าที่หลัก



ประการแรก ๆ ของ shell ตัวนี้ก็คือการจัดส่ง DOS requests ต่าง ๆ ไปให้เซอร์ฟเวอร์ นอกจากนี้ก็จะให้บริการอื่น เช่น NetWare-aware เพื่อสนับสนุนแอพพลิเคชั่นต่าง ๆ ด้วย function calls เพื่อเรียกใช้บริการต่าง ๆ ที่เรียกว่า client services ได้ โดยอาศัยโปรโตคอลของเน็ตแวร์ที่เรียกว่า NetWare Core Protocol (NCP) ที่มีขีดความสามารถเหนือผลิตภัณฑ์อื่น ๆ ในแง่ของการเป็น advanced protocol อย่างหนึ่ง

ก่อนที่ NCP จะผ่านมาที่เซอร์ฟเวอร์โดยทาง communications-subsystem นั้น ตัว NCP เองจะต้องไปรับ process ที่ shell เสียก่อนเป็นอันดับแรก ความต้องการใช้งานต่าง ๆ ที่ผู้ใช้เรียกเข้ามาที่เซอร์ฟเวอร์นั้นจะเรียกว่า request-server-processes กระบวนการที่เป็นกระดูกสันหลังของสภาพแวดล้อมของเซอร์ฟเวอร์ในเน็ตแวร์ ซึ่งมีหน้าที่บริหารจัดการเกี่ยวกับ file system accesses ต่าง ๆ ตลอดจนเรื่องของ print jobs, routing, security และหน้าที่สำคัญอื่น ๆ

โปรแกรมมิ่งภายใต้เน็ตแวร์

OS ของเน็ตแวร์นั้นไม่ใช่ DOS และไม่ได้พัฒนามาจาก DOS แต่อย่างใด จึงมีผู้เป็นห่วงในเรื่องของคอมแพตทิบิลิตระหว่างเน็ตแวร์กับ DOS อย่างมาก ซึ่งได้มีการปรับปรุงในเรื่องนี้อยู่แล้วในส่วนที่เป็น DOS ในแง่ของการขยายขีดความสามารถของ file-access calls ต่าง ๆ ใน DOS ให้สามารถสนับสนุนสภาพแวดล้อมที่เป็นเน็ตเวอร์กได้ด้วย

NetWare 2.1 สนับสนุน DOS INT 21H functions ทุกอันอย่างครบครัน ซึ่งรวมถึง extended file opens, older compatibility opens และ file-locking operations ต่าง ๆ โดยที่การทำงานในโหมดที่เกี่ยวกับ extended file opens นั้นจะต้องมาก่อนเรื่องของ Novell sharing scheme เช่น ถ้าได้มีการกำหนดให้ Novell file อันใดอันหนึ่งเป็น non-shareable เมื่อใด ผู้ใช้งานหลาย ๆ รายก็จะสามารถเปิดไฟล์อันนี้ได้โดยอาศัย DOS nonexclusive open ได้พร้อม ๆ กัน แต่ถ้าการใช้งานอันหนึ่งไปเปิดไฟล์ที่เป็น Novell shareable ประเภท deny-read write เมื่อใด ก็จะมีผลให้การ opens อื่น ๆ ที่ตามมาต้องถูกปฏิเสธเมื่อนั้น

นอกจากนี้เน็ตแวร์ยังให้การสนับสนุนการทำหน้าที่ต่าง ๆ ในด้าน data management อีกด้วย เช่น ผู้ใช้จะสามารถล็อคเรคอร์ดต่าง ๆ ได้ด้วยการกำหนดค่าของ byte-offset หรืออาศัย logical names การสนับสนุนอื่นก็มี เช่น การเข้าถึงไฟล์ในฐานข้อมูลโดยอาศัยระบบที่เรียกว่า transaction tracking system (TTS) ที่มองลำดับการปฏิบัติงานของฐานข้อมูลที่มีหลาย ๆ ขั้นตอนให้เหลือเป็นรายการเดียว (single transaction) ฯลฯ

การสนับสนุนอื่นก็มีอีก เช่น extended function calls ต่าง ๆ สำหรับ DOS file operations รวมทั้ง locking calls ที่จะเป็นประโยชน์กับโปรแกรมเมอร์ ซึ่งเมื่อรวมทั้งหมดแล้วจะมี NCP functions ให้เลือกใช้ได้ไม่น้อยกว่า 150 แบบ ในแง่ของการทำหน้าที่เป็น client server interface สำหรับการใช้งานต่าง ๆ บน client stations ทันทีที่ได้มีการโหลดตัว shell เข้าไป อินเตอร์เฟสที่สำคัญอีกอันก็มี application programming interfaces (API) ที่เน็ตแวร์ให้การสนับสนุน ซึ่งจะมีทั้งหมด 20 APIs ด้วยกัน รวมทั้ง database API ที่ใช้ Softcraft's Btrieve ให้เป็น record manager บนเซอร์ฟเวอร์ ส่วนคำว่า "Value added processes" (VAP) ที่พูดถึงในรูปที่ 5 นั้นจะหมายถึง third-party applications ที่วิ่งอยู่ในเซอร์ฟเวอร์ของเน็ตแวร์ API supports ที่น่ากล่าวถึงอีกอันหนึ่งก็คือ เรื่องของ job-queuing ที่จะมาสนับสนุนการพัฒนาใช้งานต่าง ๆ บนเซอร์ฟเวอร์สำหรับ batch jobs, shared printing, large compiles และงานอื่น ๆ ที่ต้องการการจัดลำดับในเซอร์ฟเวอร์ของเน็ตแวร์

การสนับสนุนปลีกย่อยอื่น ๆ ของเน็ตแวร์ก็มี อาทิเช่น network resource monitoring และ diagnostic functions ต่าง ๆ โดยที่ API ที่ทำหน้าที่ในด้าน network management และ diagnostic จะเปิดโอกาสให้โปรแกรมเก็บสถิติจาก IPX/SPX, ระบบดิสก์, TTS, login connection tables และอื่น ๆ ได้

สรุป

เน็ตแวร์นั้นมีสรรพคุณข้อดีดังได้กล่าวมาตั้งแต่ต้นจนบัดนี้ แต่ก็ยังมีจุดอ่อนที่ต้องปรับปรุงอีกหลาย ๆ ด้าน เช่น ในแง่ของ communication services ต่าง ๆ โดยเฉพาะ modem servers กับ wide area bridges ที่ต้องการการรวมกันให้กระชับกว่าที่เป็นอยู่ในขณะนี้ เพื่อแก้ปัญหาในเรื่องของ name service ซึ่งจะมาจำกัดขีดความสามารถในเรื่อง internetworking ปัญหาอื่นของซอฟต์แวร์นี้ที่พอมองเห็นก็จะเป็นเรื่องของคอมแพตทิบิลิตกับบั๊กที่ยังหลงเหลืออยู่บ้าง แต่ก็คงมีการขจัดให้หมดไปในไม่ช้า